Robotic Information Gathering - Exploration of Unknown Environment

Jan Faigl

Department of Computer Science
Faculty of Electrical Engineering
Czech Technical University in Prague

Lecture 04

B4M36UIR – Artificial Intelligence in Robotics

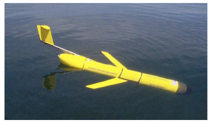
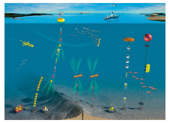
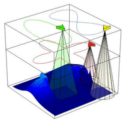
Overview of the Lecture

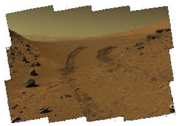
Part I

Part 1 – Robotic Exploration

Robotic Information Gathering

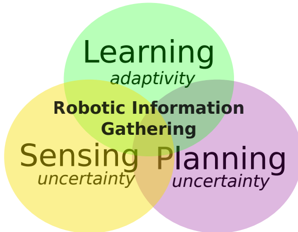
Create a model of phenomena by autonomous mobile robots performing measurements in a dynamic unknown environment.

Challenges in Robotic Information Gathering

- **Where to take new measurements?**
  To improve the phenomena model.
- **What locations visit first?**
  On-line decision-making.
- **How to efficiently utilize more robots?**
  To divide the task between the robots/
- **How to navigate robots to the selected locations?**
  Improve Localization vs Model.



How to address all these aspects altogether to find a cost-efficient solution using in–situ decisions?

Robotic Information Gathering and Multi-Goal Planning

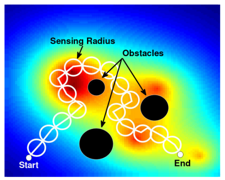
- **Robotic information gathering** aims to determine an optimal solution to collect **the most relevant data** (measurements) in a **cost-efficient way**.
  - It builds on a simple path and trajectory planning – *point-to-point planning*.
  - It may consist of determining locations to be visited and a combinatorial optimization problem to determine the sequence to visit the locations.
- It can be considered a general problem for various tasks and missions, including **online decision-making**.
  - Informative path/motion planning and persistent monitoring.
  - **Robotic exploration** – create a map of the environment as quickly as possible.

  and **determining a plan** according to the particular **assumptions and constraints**; a plan that is then executed by the robots.
  - **Inspection planning** - Find a shortest tour to inspect the given environment.
  - **Surveillance planning** - Find the shortest (a cost-efficient) tour to periodically monitor/capture the given objects/regions of interest.
  - **Data collection planning** – Determine a cost-efficient path to collect data from the sensor stations (locations).
- In both cases, **multi-goal path planning** allows solving (or improving the performance) of the particular missions.

Informative Motion Planning

- **Robotic information gathering** can be considered as the **informative path planning** problem to a determine trajectory $\mathcal{P}^*$ such that

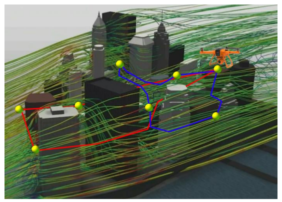
  $\mathcal{P}^* = \text{argmax}_{\mathcal{P}\in\Psi} I(\mathcal{P})$, such that $c(\mathcal{P}) \le B$, where
  - Ψ is the space of all possible robot trajectories,
  - $I(\mathcal{P})$ is the information gathered along the trajectory $\mathcal{P}$,
  - $c(\mathcal{P})$ is the cost of $\mathcal{P}$ and B is the allowed budget.
- Searching the space of all possible trajectories is complex and demanding problem.
- A discretized problem can be solved by combinatorial optimization techniques. *Usually scale poorly with the size of the problem.*
- A trajectory is from a continuous domain.
- **Sampling-based path/motion planning techniques** can be employed for finding maximally informative trajectories.



Hollinger, G., Sukhatme, G. (2014): Sampling-based robotic information gathering algorithms. IJRR.

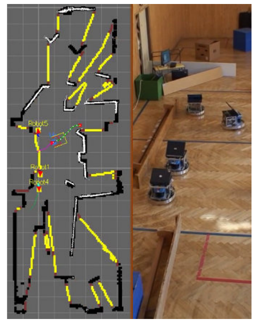
Persistent Monitoring of Spatiotemporal Phenomena

- Persistent environment monitoring is an example of the robotic information gathering mission.
- It stands to determine suitable locations to collect data about the studied phenomenon.
- Determine a cost-efficient path to visit the locations, e.g., considering a limited travel budget. *Orienteering Problem*
- Collect data and update the phenomenon model.
- Search for the next locations to improve the model.
- **Robotic information gathering** is challenging problem.
  - **Optimal sampling design** to Determine locations to be visited w.r.t. the mission objective.
  - **Trajectory planning** – **Path/motion planning** to find optimal paths/trajectories.
  - **Multi-goal path/motion planning** for an optimal sequence of visits to the locations.
  - Solutions have to respect, e.g., kinematic and kinodynamic constraints, collision-free paths.
    In general, the problem is very challenging, and therefore, we consider the most important and relevant constraints, i.e., we address the problem under particular assumptions.



Robotic Exploration of Unknown Environment

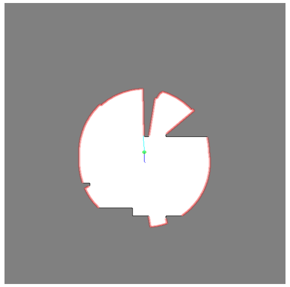
- Robotic exploration is a fundamental problem of robotic information gathering.
  How to efficiently utilize a group of mobile robots to create a map of an unknown environment autonomously?
- Performance indicators vs. constraints.
  - **Indicators** – time, energy, map quality.
  - **Constraints** – no. of robots, communication.
- Performance in a real mission depends on the on-line **decision-making**.
- It includes multiple challenges:
  - Map building and localization;
  - Determination of the navigational waypoints; *Where to go next?*
  - Path planning and navigation to the waypoints;
  - Coordination of the actions (multi-robot team).



Courtesy of M. Kulich

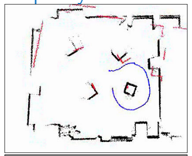
Mobile Robot Exploration

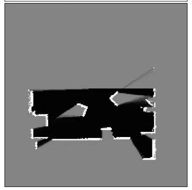
- Create a map of the environment.
- **Frontier**-based approach. *Yamauchi (1997)*
- Occupancy grid map. *Moravec and Elfes (1985)*
- Laser scanner sensor.
- Next-best-view approach. *Select the next robot goal*
- Performance metric, e.g.,
  Time to create a map of the whole environment
  vs. time to search entity in a search-and-rescue mission.



Environment Representation – Mapping and Occupancy Grid

- The robot uses its sensors to build a map of the environment.
- The robot should be localized to integrate new sensor measurements into a globally consistent map.
- **Simultaneous Localization and Mapping** (SLAM).
  - The robot uses the map being built to localize itself.
  - The map is primarily to help to localize the robot.
  - The map is a "side product" of SLAM.
- **Grid map** – discretized world representation.
  - A cell is occupied (an obstacle) or free.
- **Occupancy grid map** – Each cell is a binary random variable modeling the occupancy of the cell.





Courtesy of M. Kulich

Occupancy Grid

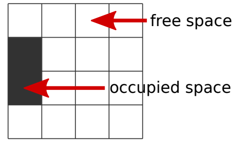
- **Assumptions**
  - The area of a cell is either completely free or occupied.
  - Cells (random variables) are independent of each other.
  - The state is static.
- A cell is a binary random variable modeling the occupancy of the cell, e.g.,
  - Cell m_i is occupied $p(m_i) = 1$;
  - Cell m_i is not occupied $p(m_i) = 0$;
  - Unknown $p(m_i) = 0.5$.
- Probability distribution of the map m

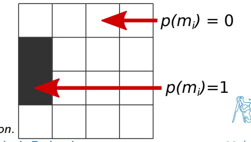
$$p(m) = \Pi_i p(m_i).$$

- Estimation of the map from sensor data $z_{1:t}$ and robot poses $x_{1:t}$

$$p(m|z_{1:t}, x_{1:t}) = \Pi_i p(m_i|z_{1:t}, x_{1:t}).$$

Binary Bayes filter – Bayes rule and Markov process assumption.





Binary Bayes Filter

- Sensor data $z_{1:t}$ and robot poses $x_{1:t}$.
- Binary random variables are independent and states are static.

$$p(m_i|z_{1:t}, x_{1:t}) \overset{\text{Bayes rule}}{=} \frac{p(z_t|m_i, z_{1:t-1}, x_{1:t})p(m_i|z_{1:t-1}, x_{1:t})}{p(z_t|z_{1:t-1}, x_{1:t})} \overset{\text{Markov}}{=} \frac{p(z_t|m_i, x_t)p(m_i|z_{1:t-1}, x_{1:t-1})}{p(z_t|z_{1:t-1}, x_{1:t})}$$

$$p(z_t|m_i, x_t) = \frac{p(m_i, z_t, x_t)p(z_t, x_t)}{p(m_i|x_t)}$$

$$p(m_i, z_{1:t}, x_{1:t}) \overset{\text{Bayes rule}}{=} \frac{p(m_i|z_t, x_t)p(z_t|x_t)p(m_i|z_{1:t-1}, x_{1:t-1})}{p(m_i|x_t)p(z_t|z_{1:t-1}, x_{1:t})} \overset{\text{Markov}}{=} \frac{p(m_i|z_t, x_t)p(z_t|x_t)p(m_i|z_{1:t-1}, x_{1:t-1})}{p(m_i)p(z_t|z_{1:t-1}, x_{1:t})}$$

- Probability a cell is occupied

$$p(m_i|z_{1:t}, x_{1:t}) = \frac{p(m_i|z_t, x_t)p(z_t|x_t)p(m_i|z_{1:t-1}, x_{1:t-1})}{p(m_i)p(z_t|z_{1:t-1}, x_{1:t})}$$

- Probability a cell is not occupied

$$p(\neg m_i|z_{1:t}, x_{1:t}) = \frac{p(\neg m_i|z_t, x_t)p(z_t|x_t)p(\neg m_i|z_{1:t-1}, x_{1:t-1})}{p(\neg m_i)p(z_t|z_{1:t-1}, x_{1:t})}$$

- Ratio of the probabilities

$$\frac{p(m_i|z_{1:t}, x_{1:t})}{p(\neg m_i|z_{1:t}, x_{1:t})} = \frac{p(m_i|z_t, x_t)p(m_i|z_{1:t-1}, x_{1:t-1})p(\neg m_i)}{p(\neg m_i|z_t, x_t)p(\neg m_i|z_{1:t-1}, x_{1:t-1})p(m_i)} = \frac{p(m_i|z_t, x_t)}{1 - p(m_i|z_t, x_t)} \frac{p(m_i, z_{1:t-1}, x_{1:t-1})}{1 - p(m_i|z_{1:t-1}, x_{1:t-1})} \frac{1 - p(m_i)}{p(m_i)}.$$

sensor model z_t, recursive term, prior

- Log odds ratio is defined as $l(x) = \log \frac{p(x)}{1-p(x)}$ and the probability $p(x)$ is $p(x) = 1 - \frac{1}{1-e^{l(x)}}$.
- The product modeling the cell m_i based on $z_{1:t}$ and $x_{1:t}$.

$$l(m_i|z_{1:t}, x_{1:t}) = \underbrace{l(m_i|z_t, x_t)}_{\text{inverse sensor model}} + \underbrace{l(m_i, |z_{1:t-1}, x_{1:t-1})}_{\text{recursive term}} - \underbrace{l(m_i)}_{\text{prior}}$$

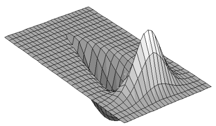
Occupancy Mapping Algorithm

Algorithm 1: OccupancyGridMapping($\{l_{t-1,i}\}$,x_t, z_t)

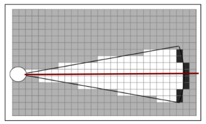
foreach *m_i of the map m* **do**
  if *m_i in the perceptual field of z_t* **then**
    $l_{t,i} := l_{t-1,i} + \text{inv\_sensor\_model}(m_i, x_t, z_t) - l_0$;
  else
    $l_{t,i} := l_{t-1,i}$;

return $\{l_{t,i}\}$

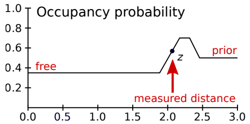
- Occupancy grid mapping has been developed by Moravec and Elfes in mid 80'ies for noisy sonars.



Inverse sensor model for sonars range sensors



Field of view of the sonar range sensor



Occupancy value depending on the measured distance

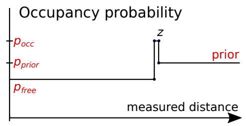
Laser Sensor Model

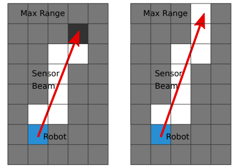
- The model is "sharp" with the precise obstacle detection.
- For the range measurement d_i, update the grid cells along a sensor beam, e.g., using Bresenham's algorithm.

Algorithm 2: Update map for $\mathcal{L} = (d_1, \ldots, d_n)$

foreach $d_i \in \mathcal{L}$ **do**
  foreach *cell m_i raycasted towards* $\min(d_i, range)$ **do**
    $p := grid(m_i)p_{free}$;
    $grid(m_i) := p/(2p - p_{free} - grid(m_i) + 1)$;
  m_d := cell at d_i;
  if *obstacle detected at m_d* **then**
    $p := grid(m_d)p_{occ}$;
    $grid(m_i) := p/(2p - p_{occ} - grid(m_i) + 1)$
  else
    $p := grid(m_d)p_{free}$;
    $grid(m_i) := p/(2p - p_{free} - grid(m_i) + 1)$

- Multiple cells can be updated by beam raycasting.



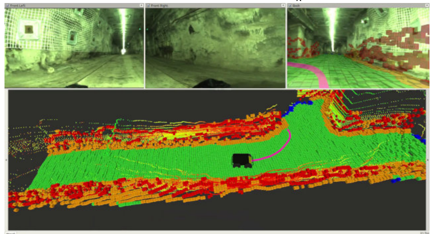


J. Amanatides and A. Woo (1987), A Fast Voxel Traversal Algorithm for Ray Tracing, Eurographics.
X. Wu (1991), An Efficient Antialiasing Technique, SIGGRAPH Computer Graphics.
C. Schulz and A. Zell (2019), Sub-Pixel Resolution Techniques for Ray Casting in Low-Resolution Occupancy Grid Maps, ECMR.

2.5D Environment Representation – Elevation Map

- An extension of the 2*D* occupancy map to 2.5D elevation map, where each cell includes information about the terrain elevation, e.g., using Kalman filter update for the elevation h after observation z_k.

$$h_k = \frac{\sigma_k^2 h_{k-1} + \sigma_{k-1}^2 z_k}{\sigma_k^2 + \sigma_{k-1}^2} \quad \sigma_k^2 = \frac{\sigma_k^2 \sigma_{k-1}^2}{\sigma_k^2 + \sigma_{k-1}^2}$$



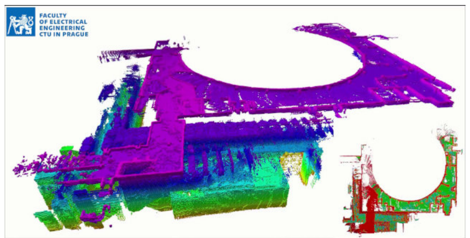
Bayer, J. and Faigl, J.: *Speeded Up Elevation Map for Exploration of Large-Scale Subterranean Environments*, 2019 Modelling and Simulation for Autonomous Systems (MESAS), 2020, pp. 190-202.

3D Occupancy Grid Environment Representation – OctoMap

- The idea of the occupancy grid can be extended to 3D using octrees – OctoMap.

`https://octomap.github.io/`, `http://wiki.ros.org/octomap`

Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., and Burgard, W. 2013, *Octomap: An Efficient Probabilistic 3d Mapping Framework Based on Octrees*, Autonomous Robots, 34:189–206.



Courtesy of the CTU-CRAS-NORLAB team, 2020 – `https://robotics.fel.cvut.cz/cras/darpa-subt/`

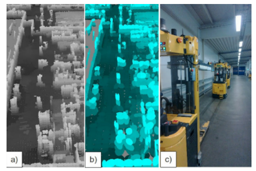
Environment Representation: Unbound by Resolution

- **Normal Distribution Transform** Occupancy Map (NDT-OM)
  - Each cell is described by a (set of) normal distribution(s).

  Saarinen, J., Andreasson, H., Stoyanov, T., Ala-Luhtala, J., Lilienthal, A.J.: *Normal Distributions Transform Occupancy Maps: Application to large-scale online 3D mapping*, ICRA, 2013.
- **Gaussian Processes** (GPs) might model occupancy or elevation as a function of position - fill in gaps between measurements.
  - Gaussian Process predicts a normal distribution - description of prediction uncertainty.

  Vasudevan, S., Ramos, F., Nettleton, E., Durrant-Whyte, H., Blair, A.: *Gaussian Process Modeling of Large Scale Terrain*, ICRA, 2009.
  Ruiz, A.V., Olariu, C.: *A General Algorithm for Explorationwith Gaussian Processes in Complex, Unknown Environments*, ICRA, 2015.
- **Gaussian Mixture Models** (GMMs) can model observed surfaces.

  O'Meadhra, C., Tabib, W., Michael, N.: *Variable Resolution Occupancy Mapping usingGaussian Mixture Models*, IEEE Robotics and Automation Letters, 2019.
  Tabib, W., Goel, K., Yao, John, Dabhi, M., Boirum, C., Michael, N.: *Real-Time Information-Theoretic Explorationwith Gaussian Mixture Model Map*, RSS, 2019.

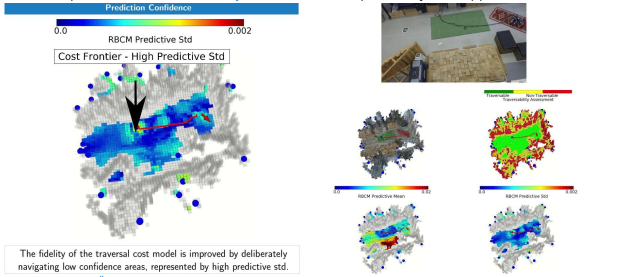


a-NDT-OM, b-low resolution map, c-real scene

Elevation map generated by neural network GP

Kriging in Spatial Modeling

- The robot can build a model of phenomena underlying the spatial model, such as pollution, radiation, temperature, or traversability assessment in a previously unmapped environment.

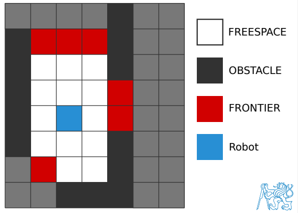


Prágr, Čížek, Bayer, Faigl: *Online Incremental Learning of the Terrain Traversal Cost in Autonomous Exploration*, Robotics: Science and Systems (RSS), 2019.

Frontier-based Exploration

- The basic idea of the **frontier** based exploration is a navigation of the mobile robot towards unknown regions. Yamauchi: *A frontier-based approach for autonomous exploration*, **CIRA** 1997.
- **Frontier** – a border of the known free space and unknown regions of the environment.
- Based on the probability of individual cells in the occupancy grid, cells are classified into three classes, e.g.,
  - FREESPACE: $p(m_i) < 0.4$;
  - UNKNOWN: $0.4 \le p(m_i) \le 0.6$;
  - OBSTACLE: $p(m_i) > 0.6$.
- **Frontier cell** is a FREESPACE cell that is incident with an UNKNOWN cell.
- Frontier cells as the navigation waypoints have to be reachable, e.g., after obstacle growing.

  Use grid-based path planning



Frontier-based Exploration Strategy

Algorithm 3: Frontier-based Exploration

map := init(*robot*, *scan*);
while *there are some reachable frontiers* **do**
  Update occupancy *map* using new sensor data and Bayes rule;
  $\mathcal{M}$:= Created grid map from *map* using thresholding;
  $\mathcal{M}$:= Grow obstacle according to the dimension of the robot;
  $\mathcal{F}$:= Determine frontier cells from $\mathcal{M}$;
  $\mathcal{F}$:= Filter out unreachable frontiers from $\mathcal{F}$;
  f := Select the closest frontier from $\mathcal{F}$, e.g. using shortest path;
  path := Plan a path from the current robot position to f;
  Navigate robot towards f along *path* (for a while);

- Exploration is an iterative decision-making process with simultaneous localization and mapping running in parallel.
- Based on the current map of the environment, new goals location candidates are generated from the frontier cells.
- Candidate locations are examined, and the "most suitable" (closest) goal (frontier cell) is selected as a new goal location.

  Path planning is performed during the examination of candidates.
- The robot is navigated towards the goal until the "replanning" is triggered.

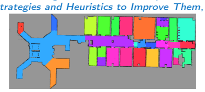
Improvements of the basic Frontier-based Exploration

Several improvements have been proposed in the literature

- Introducing utility based on the expected covered area from a particular location (frontier cell).

  González-Baños, Latombe: *Navigation Strategies for Exploring Indoor Environments*, IJRR, 2012.
- Map segmentation for identification of rooms and exploration of the whole room by a single robot.

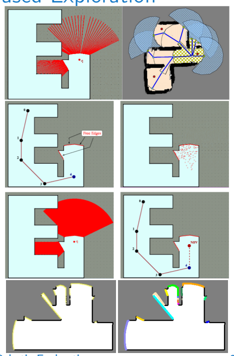
  Holz, Basilico, Amigoni, Behnke: *A Comparative Evaluation of Exploration Strategies and Heuristics to Improve Them*, ECMR, 2011.



- Consider a longer planning horizon as a solution to the **Traveling Salesman Problem** (**TSP**).

  Zlot, Stentz (2006), Kulich, Faigl (2011, 2012)
- **Representatives of free edges** – Frontier cells are formed into connected components that represent the free edges.

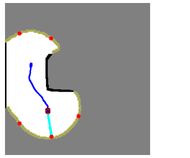
  Kulich, Faigl (2011, 2013)



Variants of the Distance Cost

- **Simple robot-goal distance** – *next-best view.*
  - Evaluate all goals using the robot–goal distance.

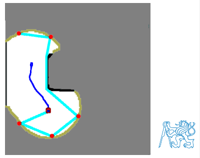
    A length of the path from the robot position to the goal candidate.
  - Greedy goal selection – the closest one.
  - Using frontier representatives improves the performance a bit.



- **TSP distance cost** – *Non-myopic next-best view.*
  - Consider visitations of all goals.

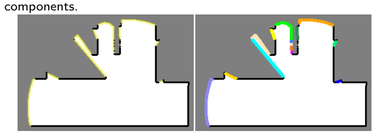
    Solve the associated traveling salesman problem (TSP).
  - A length of the tour visiting all goals.
  - Use frontier representatives – to avoid large instances of the TSP.
  - the TSP distance cost improves performance about 10–30% without further heuristics, e.g., expected coverage (utility).

  Kulich, M., Faigl, J, Přeučil, L.: *On Distance Utility in the Exploration Task*, ICRA, 2011.



Frontier Representatives – Frontier Clusters

- An omnidirectional sensor with a non-zero sensing range can cover multiple frontier cells.
- Group frontier cells to the so-called **free-edges** – single connected components.

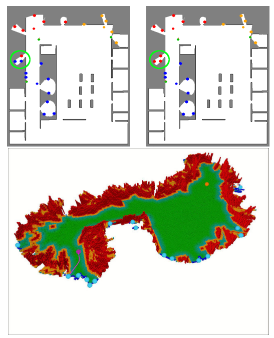


- Split large clusters (of the size f) to smaller clusters that can be covered by the sensor range D; determine the number of subclusters n_r and use **k-means** clustering.

$$n_r = 1 + \left\lfloor \frac{f}{1.8D} + 0.5 \right\rfloor .$$

Faigl, J., Kulich, M., and Přeučil, L.: *Goal assignment using distance cost in multi-robot exploration*, IROS 2012.

- It reduces the number of goal candidates and yields navigation towards middle locations of the free-edges.



Multi-robot Exploration

- **Multi-robot exploration** is a problem to efficiently utilize a **group** of (mobile) robots to autonomously create a model of a priory unknown environment.
- **Uncoordinated** approach – Each robot independently explores the environment, e.g., by following the closest frontier.
- **Centralized** approaches – a central authority assigns the goals, and the goal assignment can be viewed as the **task allocation problem**.
  - Various strategies have been proposed, such as greedy assignment, Hungarian assignment, and multiple traveling salesman problem assignments.

    Considering communication between the exploring units, we can further establish distributed task allocation.
- **Decentralized** approaches – Each robot selects its own goal and solves the task allocation based on its (limited) information about other robots.

  Existing communication between the exploring units can improve the performance, but it is generally not mandatory for "true" decentralized approaches.

Multi-robot Exploration – Overview of Centralized Strategy

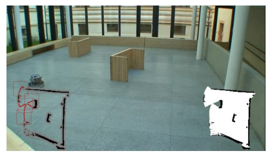
- We need to assign navigation waypoint to each robot that can be formulated as the **task-allocation problem**.
- Multi-robot exploration as an **iterative procedure**.
  1. Initialize the occupancy grid Occ.
  2. $\mathcal{M} \leftarrow$ create_navigation_grid(Occ).

     cells of $\mathcal{M}$ *have values* $\{freespace, obstacle, unknown\}$.
  3. $\boldsymbol{F} \leftarrow$ detect_frontiers($\mathcal{M}$).
  4. Goal candidates $\boldsymbol{G} \leftarrow$ generate($\boldsymbol{F}$).
  5. **Assign next goals to each robot** $r \in \boldsymbol{R}$,

     $(\langle r_1, g_{r_1}\rangle, \ldots, \langle r_m, g_{r_m}\rangle) = \text{assign}(\boldsymbol{R}, \boldsymbol{G}, \mathcal{M})$.
  6. Create a plan $\boldsymbol{P_i}$ for each pair $\langle r_i, g_{r_i}\rangle$.

     consisting of simple operations.
  7. Perform each plan up to s_{max} operations.

     At each step, update Occ using new sensor measurements.
  8. If $|G| == 0$ exploration finished, otherwise go to Step 2.



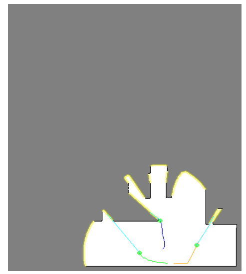
- Several parts of the exploration procedure are important regarding decision-making and achieved performance.
  - How to determine goal candidates from the the frontiers?
  - How to plan a paths and assign the goals to the robots?
  - How to navigate the robots towards the goal?
  - When to replan?

Exploration Procedure – Decision-Making Parts

1. Initialize – set of plans for m robots, $\mathcal{P} = (P_1, \ldots, P_m)$, $P_i = \emptyset$.
2. Repeat
   2.1 **Navigate robots** using the plans $\mathcal{P}$;
   2.2 Collect new measurements;
   2.3 Update the navigation map $\mathcal{M}$;

   Until **replanning condition is met**.
3. **Determine goal candidates** $\boldsymbol{G}$ **from** $\mathcal{M}$.
4. If $|\boldsymbol{G}| > 0$ **assign goals to the robots**
   - $(\langle r_1, g_{r_1}\rangle, \ldots, \langle r_m, g_{r_m}\rangle)$=assign($\boldsymbol{R}, \boldsymbol{G}, \mathcal{M}$), $r_i \in \boldsymbol{R}, g_{r_i} \in \boldsymbol{G}$;
   - **Plan paths** to the assigned goals $\mathcal{P} = \text{plan}(\langle r_1, g_{r_1}\rangle, \ldots, \langle r_m, g_{r_m}\rangle, \mathcal{M})$;
   - Go to Step 2.
5. Stop all robots or navigate them to the depot.

   All reachable parts of the environment are explored.



Determination of goal locations and path (cost) to them.

Goal Assignment Strategies – Task Allocation Algorithms

- Exploration strategy can be formulated as the **task-allocation problem**

$$(\langle r_1, g_{r_1}\rangle, \ldots, \langle r_m, g_{r_m}\rangle) = \text{assign}(\boldsymbol{R}, \boldsymbol{G}(t), \mathcal{M}),$$

where $\mathcal{M}$ is the current map.

1. **Greedy Assignment**
   - Randomized greedy selection of the closest goal candidate.
   Yamauchi B., Robotics and Autonomous Systems 29, 1999.
2. **Iterative Assignment**
   - Centralized variant of the broadcast of local eligibility algorithm (BLE).
   Werger, B., Mataric, M., Distributed Autonomous Robotic Systems 4, 2001
3. **Hungarian Assignment**
   - Optimal solution of the task-allocation problem for assignment of n goals and m robots in $O(n^3)$. For $n < m$: use Iterative assignment or dummy tasks; For $n > m$: add dummy robots with costly assignments.
   Stachniss, C., C implementation of the Hungarian method, 2004
4. **Multiple Traveling Salesman Problem – MTSP Assignment**
   - ⟨cluster–first, route–second⟩, the TSP distance cost. Faigl, et al. 2012

MTSP-based Task-Allocation Approach

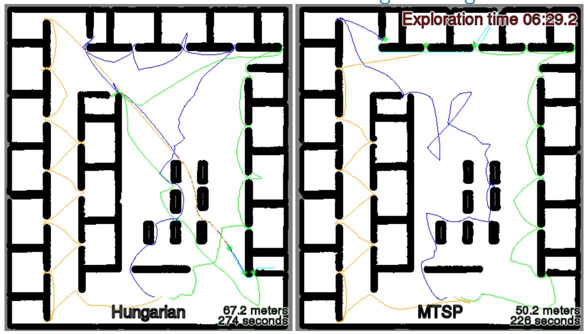
- Task-allocation problem as the **Multiple Traveling Salesman Problem (MTSP)**.
- m-TSP heuristic *⟨cluster–first, route–second⟩*
  1. Cluster the goal candidates $\boldsymbol{G}$ to m clusters (using k-means)

  $$\boldsymbol{C} = \{C_1, \ldots, C_m\}, C_i \subseteq \boldsymbol{G}.$$

  2. For each robot $r_i \in \boldsymbol{R}, i \in \{1, \ldots m\}$ select the next goal g_i from C_i using **the TSP distance cost**. *Kulich et al., ICRA (2011)*
     - Solve the TSP on the set $C_i \cup \{r_i\}$ – the tour starts at r_i.
     - The next robot goal g_i is the first goal of the found TSP tour.

Faigl, J., Kulich, M., Přeučil, L.: *Goal Assignment using Distance Cost in Multi-Robot Exploration*, **IROS** 2012.

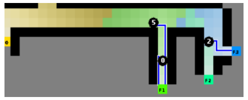
Performance of the MTSP vs Hungarian Algorithm



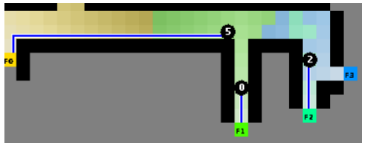
Replanning as quickly as possible; $m = 3$, $\rho = 3$ m – *The MTSP assignment provides better performance.*

MinPos: Decentralized Exporation Strategy

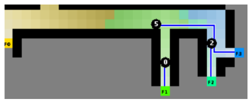
- The robot solves the task allocation based on its (limited) information about other robots.
- **Assumption**: the distance cost matrix C between robots $\mathcal{R}$ and frontiers $\mathcal{F}$ are known to all robots. *In practice, it requires the robots to share the map of the whole environment, which might not be feasible, and therefore, approximations can be employed.*
- Each robot **ranks** each frontier using the relative distance of the robots to the frontier cell (goal candidate).
- The robot is assigned the goal with the minimum rank.



Nearest goal candidate (frontier) selection



Minpos assignment

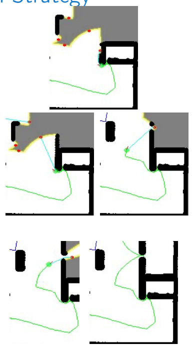


Gready assignment of goal candidates (frontiers)

Bautin, A., Simonin, O., Charpillet, F.: *MinPos: A Novel Frontier Allocation Algorithmfor Multi-robot Exploration*, ICIRA, 2012.
Faigl, J., Simonin, O., Charpillet, F.: *Comparison of Task-Allocation Algorithms in Frontier-Based Multi-robot Exploration, European Conference on Multi-Agent Systems*, EUMAS, 2014.

Influence of Decision-Making – Exploration Strategy

- The exploration performance depends on the whole solution, albeit we can have "best" possible solutions of each part.
- Locally optimal Hungarian algorithm might not necessarily provide better solutions than for example the MTSP-based approach.
- A solution of the particular sub-task (i.e., **goal candidate selection**) might have side effects that are exhibited during the missions – depending on the utilized navigation technique.
  - Vector Field Histogram (VFH) slows down the robot close to the obstacles.
  Borenstein, J. and Koren, Y.: The vector field histogram-fast obstacle avoidance for mobile robots, IEEE Transactions on Robotics, 1991.
  - A side effect of the representatives of free edges is that goal candidates are "in the middle of free-edges" and the robot is navigated towards them, which results in faster motion because it is relatively far from the obstacles.



It is all related to simplifications we made to solve the challenging autonomous exploration.

Information Theory in Robotic Information Gathering

- Frontier-based exploration assumes perfect knowledge about the robot states and the utility function depends only on the map.
- We can avoid such assumption by defining the **control policy** as a rule how to select the robot action that reduces the uncertainty of estimate by learning measurements:

$$\text{argmax}_{a \in A} I_{MI}[x; z|a],$$

where A is a set of possible actions, x is a future estimate, and z is future measurement
- **Mutual information** – how much uncertainty of x will be reduced by learning z

$$I_{MI}[x; z] = H[x] - H[x|z],$$

where $H[x]$ is the current **entropy**, and $H[x|z]$ is future/**predicted entropy**.
- **Conditional Entropy** $H[x|z]$ is the expected uncertainty of x after learning unknown z (collecting new measurements).
- **Entropy** – uncertainty of x: $H[x] = -\int p(x) \log p(x) dx$.

Computing Mutual Information in Exploration

- Sensor placement approach with raycasting of the sensor beam and determination of the distribution over the range returns.
- Precise computing of the mutual information is usually not computationally feasible given the size of the action set and the uncertainty of action results.
- We can assume that observation removes all uncertainty from observed areas

$$I_{MI}[x; z] = H[x] - H[x|z] \approx H[x].$$

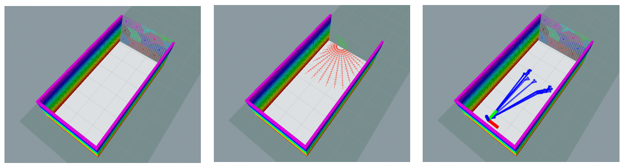
  - Then, we can decrease the computational requirements by using simplified approach where the action is selected to maximize the entropy over the sensed regions in the current map.
  - We are maximizing mutual information in the **sensor placement problem** of observing the region with maximum entropy

  $$\text{argmax}_{a \in A} \sum_{x \in \boldsymbol{R}(a)} H[p(x)],$$

  where $\boldsymbol{R}(a)$ represents the region sensed by the action a.
  Bourgault, F., Makarenko, A.a., Williams, S.B., Grocholsky, B., Durrant-Whyte, H.F.: *Information based adaptive robotic exploration*, IROS, 2002.
- Computational cost can be decreased using Cauchy-Schwarz Quadratic Mutual Information (CSQMI) defined similarly to mutual information. Can be evaluated analytically for occupancy grid mapping.
Charrow, B., Liu, S., Kumar, V., Michael, N.: *Information-theoretic mapping using Cauchy-Schwarz Quadratic Mutual Information*, ICRA 2015.

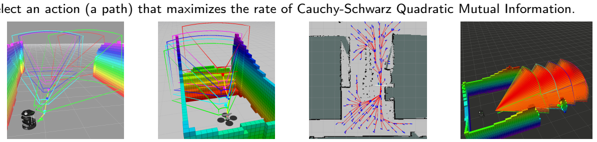
Actions

- Actions are shortest paths to cover the frontiers.

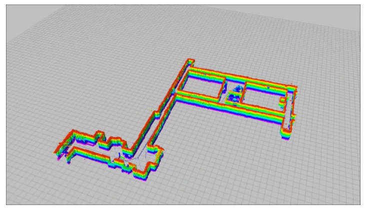


Detect and cluster frontiers Sampled poses to cover a cluster Paths to the sampled poses

- Select an action (a path) that maximizes the rate of Cauchy-Schwarz Quadratic Mutual Information.



Example of Autonomous Exploration using CSQMI



Ground vehicle

Aerial vehicle

- Planning with trajectory optimization – determine trajectory maximizing I_{CS}.

Charrow, B., Kahn, G., Patil, S., Liu, S., Goldberg, K., Abbeel, P., Michael, N., Kumar, V.: *Information-Theoretic Planning with Trajectory Optimization for Dense 3D Mapping*. **Robotics: Science and Systems (RSS)**, 2015.

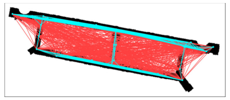
Mutual Information in Kriging

- The GP regressors provide an **inbuilt representation of uncertainty** – their prediction is a normal distribution.
  - The differential entropy of a normal distribution is

$$H(\mathcal{N}(\mu, \sigma^2)) = \frac{1}{2}\log(2\pi e \sigma^2),$$

    i.e., it is a function of its variance σ^2.
  - We can employ greedy approach - sample at the highest prediction variance.
- Example: Building communication maps
  - A pairwise problem - select locations of two robots to sample the communication signal strength.

Quattrini Li, A., Penumarthi, P.K., Banfi, J., Basilico, N., O'Kane, J.M., Rekleitis, I., Nelakuditi, S., Amigoni, F.: *Multi-robot online sensing strategies for the construction of communication maps*, Autonomous Robots 44:299—319, 2020.



Search in Kriging Scenarios

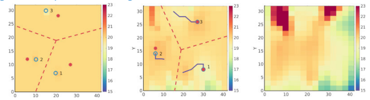
- In exploration scenarios, where we search for some phenomenon, such as searching for a source of radiation or heat, we search for the modeled function's extrema.
- The search strategy needs to **balance exploitation and exploration**.

Exploration of the current model vs. exploration of unknown parts of the environment.

- **Gaussian Process Upper Confidence Bound**
  - It addresses the search as a **multi-armed bandit** problem.
  - The GP-UCB policy to chose the next sampling point x_t is

$$x_t = \text{argmax}_{x \in D}\, \mu_{t-1}(x) + \beta_t^{\frac{1}{2}} \sigma_{t-1}(x).$$

Srinivas, N., Krause, A., Kakade, S.M., Seeger, M.: *Gaussian process optimization in the bandit setting: no regret and experimental design*, ICML 2010.



Wenhao, L., Sycara, K.: *Adaptive Sampling and Online Learning in Multi-Robot Sensor Coverage with Mixture of Gaussian Processes*, ICRA, 2018.

Exploration with Position Uncertainty

- A reliable localization is needed to map the environment reliably; thus, we might need to consider both the occupancy and localization mutual information:

$$I = \gamma I_{occupancy} + (1-\gamma) I_{localization}.$$

  - The localization uncertainty can be based on the entropy

$$\frac{1}{2}\log\left[(2\pi e)^n det P\right],$$

    where P is the covariance of location of the robot and localization landmarks.

Bourgault, F., et al.: *Information based adaptive robotic exploration*, IROS, 2002.

- Summing Shannon's entropy of the map and the differential entropy of the pose leads to scaling issues.
  - The explorer may stricly prefer to improve either its map or localization that can achieved by adjusting γ.
  - We can use the notion of Rényi's entropy

$$H_\alpha\left[P(x)\right] = \frac{1}{1-\alpha}\log_2\left(\sum p_i^\alpha\right)$$

    where for $\alpha \to 1$ its becomes Shannon's entropy.
  - The utility function of taking an action a is the difference

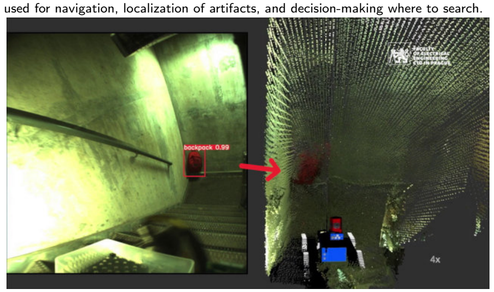
$$\text{argmax}_a \sum_{x \in R(a)} H^{\text{Shannon}}\left[P(x)\right] - H^{\text{Rényi}}_{1+\frac{1}{\delta(a)}}\left[P(x)\right]$$

    where $\delta(a)$ is related to predicted position uncertainty given the action a.

Carrillo, H., Dames, P., Kumar, V., Castellanos, J.A.: *Autonomous robotic exploration using a utility function based on Rényi's general theory of entropy*, Autonomous Robots, 2018.

Search in Unknown Environments

- A variant of exploration is a search to find objects of interest in an unknown environment.
- In search-and-rescue missions, the performance indicator is the time to find the objects and report their position.
- The **map** is used for navigation, localization of artifacts, and decision-making where to search.



Courtesy of the CTU-CRAS-NORLAB team, 2020 – `https://robotics.fel.cvut.cz/cras/darpa-subt/`

Summary of the Lecture

Topics Discussed

- Robotic information gathering – informative path planning
- Robotic exploration of unknown environment
  - Occupancy grid map
  - Frontier based exploration
  - Exploration procedure and decision-making
  - TSP-based distance cost in frontier-based exploration
  - Multi-robot exploration and task-allocation
- Mutual information and informative path planning

Motivation for the semestral project.

- Next: Multi-goal planning